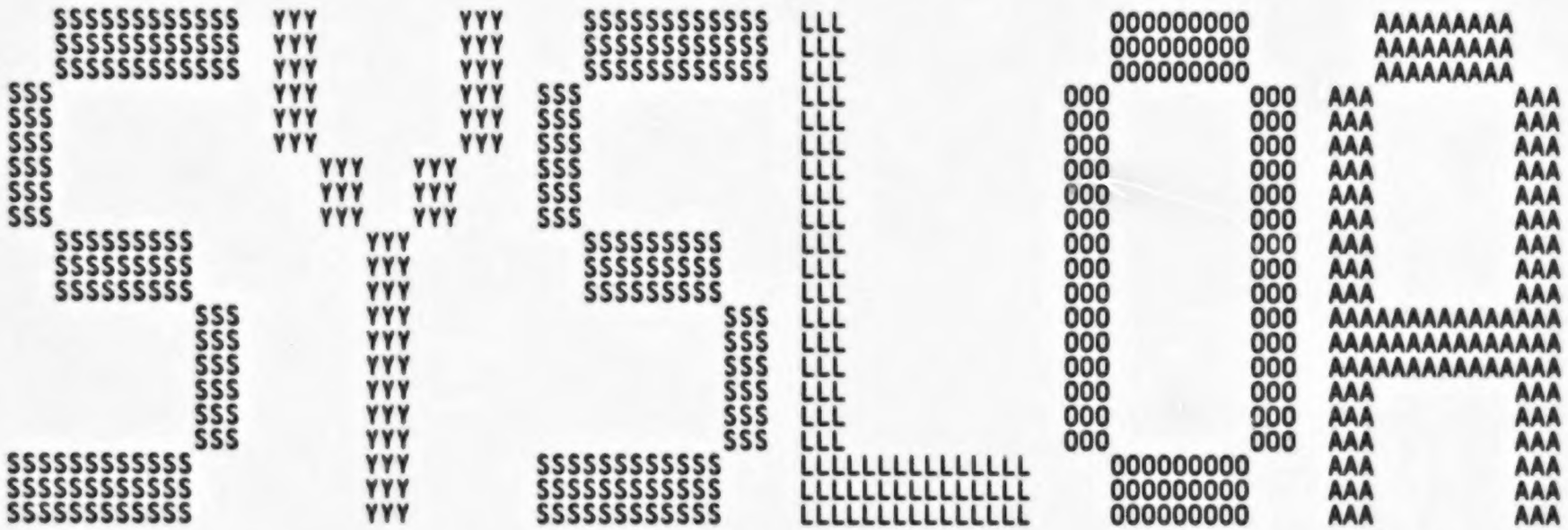
SSSSSSSSSSS YYY YYY SSSSSSSSSSS LLL OOOOOOOOO AAAAAAAAA
SSSSSSSSSSS YYY YYY SSSSSSSSSSS LLL OOOOOOOOO AAAAAAAAA
SSSSSSSSSSS YYY YYY SSSSSSSSSSS LLL OOOOOOOOO AAAAAAAAA
SSS YYY YYY SSS LLL OOO OOO AAA AAA
SSS YYY YYY SSS LLL OOO OOO AAA AAA
SSS YYY YYY SSS LLL OOO OOO AAA AAA
SSS YYY YYY SSS LLL OOO OOO AAA AAA
SSS YYY YYY SSS LLL OOO OOO AAA AAA
SSS YYY YYY SSS LLL OOO OOO AAA AAA
SSSSSSSSS YYY SSSSSSSSS LLL OOO OOO AAA AAA
SSSSSSSSS YYY SSSSSSSSS LLL OOO OOO AAA AAA
SSSSSSSSS YYY SSSSSSSSS LLL OOO OOO AAA AAA
SSS YYY SSS LLL OOO OOO AAAAAAAAAAAAAAA
SSS YYY SSS LLL OOO OOO AAAAAAAAAAAAAAA
SSS YYY SSS LLL OOO OOO AAAAAAAAAAAAAAA
SSS YYY SSS LLL OOO OOO AAA AAA
SSS YYY SSS LLL OOO OOO AAA AAA
SSSSSSSSSSS YYY SSSSSSSSSSS LLLLLLLLLLLLLLL OOOOOOOOO AAA AAA
SSSSSSSSSSS YYY SSSSSSSSSSS LLLLLLLLLLLLLLL OOOOOOOOO AAA AAA
SSSSSSSSSSS YYY SSSSSSSSSSS LLLLLLLLLLLLLLL OOOOOOOOO AAA AAA

```
  AAAAAA    DDDDDDDD    PPPPPPPP      SSSSSSSS  UU      UU  BBBBBBBB    77777777      999999      000000
  AAAAAA    DDDDDDDD    PPPPPPPP      SSSSSSSS  UU      UU  BBBBBBBB    77777777      999999      000000
AA      AA  DD      DD  PP      PP  SS          UU      UU  BB      BB          77  99      99  00      00
AA      AA  DD      DD  PP      PP  SS          UU      UU  BB      BB          77  99      99  00      00
AA      AA  DD      DD  PP      PP  SS          UU      UU  BB      BB          77  99      99  00    0000
AA      AA  DD      DD  PP      PP  SS          UU      UU  BB      BB          77  99      99  00    0000
AA      AA  DD      DD  PPPPPPPP      SSSSSS    UU      UU  BBBBBBBB        77    99999999  00  00  00
AA      AA  DD      DD  PPPPPPPP      SSSSSS    UU      UU  BBBBBBBB        77    99999999  00  00  00
AAAAAAAAAA  DD      DD  PP                  SS  UU      UU  BB      BB    77          99  0000    00
AAAAAAAAAA  DD      DD  PP                  SS  UU      UU  BB      BB    77          99  0000    00
AA      AA  DD      DD  PP                  SS  UU      UU  BB      BB  77            99  00      00
AA      AA  DD      DD  PP                  SS  UU      UU  BB      BB  77            99  00      00
AA      AA  DDDDDDDD    PP          SSSSSSSS    UUUUUUUUUU  BBBBBBBB  77          999999    000000
AA      AA  DDDDDDDD    PP          SSSSSSSS    UUUUUUUUUU  BBBBBBBB  77          999999    000000

LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

```
0000     1          .NOSHOW CONDITIONALS
0000     5
0000     9
0000    13
0000    15          .TITLE  ADPSUB790 - ADAPTER SUBROUTINES FOR VAX 11/790
0000    17
0000    21
0000    22          .IDENT  'V04-000'
0000    23
0000    24  ;******************************************************************************
0000    25  ;*                                                                            *
0000    26  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                   *
0000    27  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                     *
0000    28  ;*  ALL RIGHTS RESERVED.                                                      *
0000    29  ;*                                                                            *
0000    30  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED      *
0000    31  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE     *
0000    32  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER      *
0000    33  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY      *
0000    34  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY      *
0000    35  ;*  TRANSFERRED.                                                              *
0000    36  ;*                                                                            *
0000    37  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE      *
0000    38  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT     *
0000    39  ;*  CORPORATION.                                                              *
0000    40  ;*                                                                            *
0000    41  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
0000    42  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
0000    43  ;*                                                                            *
0000    44  ;*                                                                            *
0000    45  ;******************************************************************************
0000    46  ;
0000    47  ; Facility: System bootstrapping and initialization
0000    48  ;
0000    49  ; Abstract: This module contains initialization routines that are loaded
0000    50  ;           during system initialization (rather than linked into the system).
0000    51  ;
0000    52  ; Environment: Mode = KERNEL, Executing on INTERRUPT stack, IPL=31
0000    53  ;
0000    54  ; Author:  Kerbey T. Altmann              Creation date: 30-Oct-1982
0000    55  ;
0000    56  ; Modification history:
0000    57  ;
0000    58  ;       V03-007 TCM0002         Trudy C. Matthews       04-Jun-1984
0000    59  ;               Include more 780-specific code for the 11/790 version of
0000    60  ;               this routine.
0000    61  ;
0000    62  ;       V03-006 KPL0001         Peter Lieberwirth       12-Apr-1984
0000    63  ;               Init ADP$L_SHB properly again; V03-004 ASSUMEd this field
0000    64  ;               was at a certain constant offset, and a change to the ADP
0000    65  ;               moved it.  Note - this is a 780 change only.
0000    66  ;
0000    67  ;       V03-005 KDM0081         Kathleen D. Morse       13-Sep-1983
0000    68  ;               Create version for Micro-VAX I.
0000    69  ;
0000    70  ;       V03-004 ROW0196         Ralph O. Weber          27-JUL-1983
0000    71  ;               Correct INI$MPMADP so the ADP$L_SHB is correctly initialized
```

```
                        ;               to zero.
                        ;
                        ;       V03-003 MSH0001         Maryann Hinden          06-Dec-1982
                        ;               Add initialization for DW750.
                        ;
                        ;       V03-002 ROW0142         Ralph O. Weber          23-NOV-1982
                        ;               Correct JMP in multiport memory interrupt dispatching code
                        ;               prototype, MPMINTD, to a JSB.  MA$INT expects to receive
                        ;               control via a JSB.
                        ;
                        ;       V03-001 TCM0001         Trudy C. Matthews       8-Nov-1982
                        ;               Initialize field ADP$L_AVECTOR in INI$MPMADP.
                        ;
                        ;--
```

```
                     0000     90
00000001             0000     92          C780_LIKE = 1
                     0000     94
                     0000     98
                     0000    102
                     0000    106 ;
                     0000    107 ; MACRO LIBRARY CALLS
                     0000    108 ;
                     0000    109          $ADPDEF                         ; Define ADP offsets.
                     0000    110          $CRBDEF                         ; Define CRB offsets.
                     0000    111          $DCDEF                          ; Define AT codes.
                     0000    112          $DDBDEF                         ; Define DDB offsets.
                     0000    113          $DDTDEF                         ; Define DDT offsets.
                     0000    114          $DYNDEF                         ; Define data structure type codes.
                     0000    115          $IDBDEF                         ; Define interrupt dispatcher offsets.
                     0000    116          $MBADEF                         ; Define MASSBUS registers.
                     0000    117          $MCHKDEF                        ; Define machine check masks.
                     0000    118          $MPMDEF                         ; Define multi-port memory.
                     0000    119          $NDTDEF                         ; Define nexus device types.
                     0000    120          $PRDEF                          ; Define IPR numbers.
                     0000    121          $PTEDEF                         ; Define Page Table Entry bits.
                     0000    122          $RPBDEF                         ; Define Restart Parameter Block fields.
                     0000    123          $SSDEF                          ; Define system service codes.
                     0000    124          $UBADEF                         ; Define UBA register offsets.
                     0000    125          $UBIDEF                         ; Define UNIBUS interconnect
                     0000    126                                          ;  register offsets.
                     0000    127          $UCBDEF                         ; Define unit control block.
                     0000    128          $VADEF                          ; Define virtual address fields.
                     0000    129          $VECDEF                         ; Define vec offsets.
                     0000    130
                     0000    132          $CEBDEF                         ; COMMON EVENT BLOCK
                     0000    133          $FKBDEF                         ; FORK BLOCK
                     0000    134          $IPLDEF                         ; INTERRUPT PRIORITY LEVELS
                     0000    135          $PRIDEF                         ; PRIORITY INCREMENT DEFINITIONS
                     0000    136          $PRQDEF                         ; INTER-PROCESSOR REQUEST
                     0000    137          $RSNDEF                         ; RESOURCE NUMBER DEFINITIONS
                     0000    138          $SHBDEF                         ; SHARED MEMORY CONTROL BLOCK
                     0000    139          $SHDDEF                         ; SHARED MEMORY DATAPAGE
                     0000    141
                     0000    145
00000000             0000    146          .PSECT  SYSLOA,LONG
```

```
0000   148          .SBTTL  CI$INT - CI INTERRUPT HANDLER
0000   149 ;+
0000   150 ; CI$INT - CI INTERRUPT HANDLER
0000   151 ;
0000   152 ;       THIS MODULE IS A DUMMY CI32 INTERRUPT HANDLER WHICH IS USED
0000   153 ;       UNTIL THE REAL CI DRIVER (PADRIVER) IS LOADED. IT ALSO CONTAINS
0000   154 ;       A DUMMY CI32 CONTROLLER INITIALIZATION  ENTRY POINT.
0000   155 ;
0000   156 ; INPUTS:
0000   157 ;
0000   158 ;       THE STACK ON ENTRY IS AS FOLLOWS:
0000   159 ;
0000   160 ;               0(SP)           ADDRESS OF IDB ADDRESS
0000   161 ;       4(SP) - 16(SP)          SAVED R2 - R5
0000   162 ;               20(SP)          INTERRUPT PC
0000   163 ;               24(SP)          INTERRUPT PSL
0000   164 ;
0000   165 ; OUTPUTS:
0000   166 ;
0000   167 ;       NONE
0000   168 ;
0000   169 ; SIDE EFFECTS:
0000   170 ;
0000   171 ;       INTERRUPTS ARE DISABLED ON THE CI32
0000   172 ;-
0000   173
0000   176
0000   177 ;
0000   178 ; $PAREGDEF -- Define offsets to CI registers and fields in the registers.
0000   179 ;
0000   180
0000   181          $DEFINI PAREG
0000   182
0000   183          $DEF    PA_CNF  .BLKL   1               ; Configuration register
0004   184
0004   185          _VIELD  PA_CNF,0,<-                     ; Define config register fields:
0004   186          <ADPTYP,8,M>,-                          ;  Adapter type code
0004   187          <PFD,,M>,-                              ;  Powerfail disable
0004   188          <TDEAD,,M>,-                            ;  Transmit dead
0004   189          <TFAIL,,M>,-                            ;  Transmit fail
0004   190          <,5>,-                                  ;  5 unused bits
0004   191          <CRD,,M>,-                              ;  CRD on port init'd read
0004   192          <RDS,,M>,-                              ;  RDS on port init'd read
0004   193          <CXTER,,M>,-                            ;  SBI error confirm
0004   194          <RDTO,,M>,-                             ;  Port init'd read timeout on SBI
0004   195          <CSTMO,,M>,-                            ;  Port init'd command xmit timeout
0004   196          <,1>,-                                  ;  1 unused bit
0004   197          <PUP,,M>,-                              ;  Adapter power up
0004   198          <PDN,,M>,-                              ;  Adaptor power down
0004   199          >                                       ;
0004   200
0004   201          $DEF    PA_PMC  .BLKL   1               ; Port maint control/status register
0008   202
0008   203          _VIELD  PA_PMC,0,<-                     ; Define register fields:
0008   204          <MIN,,M>,-                              ;  Maint initialized
0008   205          <MTD,,M>,-                              ;  Maint timer disable
0008   206          <MIE,,M>,-                              ;  Maint interrupt enable
```

```
                                                0008   207          <MIF,,M>,-                              ;  Maint intterupt flag
                                                0008   208          >                                       ;
                                                0008   209
                                                0008   210          $DEFEND PAREG
                                                0000   211
                                                0000   212  CI$INT::
                    53    9E    D0            0000   213          MOVL    @(SP)+,R3                       ; GET ADDRESS OF IDB
                    54    63    D0            0003   214          MOVL    IDB$L_CSR(R3),R4                ; GET ADDRESS OF FIRST CSR
                64  00400000    8F    D0      0006   215          MOVL    #PA_CNF_M_PUP,PA_CNF(R4)        ; CLEAR POWER UP
                64  00800000    8F    D0      000D   216          MOVL    #PA_CNF_M_PDN,PA_CNF(R4)        ; CLEAR POWER DOWN
                  04 A4     01    D0          0014   217          MOVL    #PA_PMC_M_MIN,PA_PMC(R4)        ; SET MAINTENCE INITIALIZE
                    52    8E    7D          0018   218          MOVQ    (SP)+,R2                        ; RESTORE REGISTERS
                    54    8E    7D          001B   219          MOVQ    (SP)+,R4
                                02          001E   220          REI
                                            001F   221
                                            001F   224
                                            001F   225  CI$INITIAL::                                    ; CONTROLLER INITIALIZATION
                                            001F   226  CI$SHUTDOWN::                                   ; CONTROLLER SHUTDOWN
                                            001F   227
                                            001F   230
                  04 A4     01    D0          001F   231          MOVL    #PA_PMC_M_MIN,PA_PMC(R4)        ; SET MAINTENCE INITIALIZE
                                            0023   234
                                05          0023   235          RSB
```

```
                        .SBTTL  DR$INT - DR INTERRUPT HANDLER
                ;+
                ; DR$INT - DR INTERRUPT HANDLER
                ;
                ;       THIS MODULE IS A DUMMY DR32 INTERRUPT HANDLER WHICH IS USED
                ;       UNTIL THE REAL DR DRIVER (XFDRIVER) IS LOADED. IT ALSO CONTAINS
                ;       A DUMMY DR32 CONTROLLER INITIALIZATION  ENTRY POINT.
                ;
                ; INPUTS:
                ;
                ;       THE STACK ON ENTRY IS AS FOLLOWS:
                ;
                ;               0(SP)           ADDRESS OF IDB ADDRESS
                ;       4(SP) - 16(SP)          SAVED R2 - R5
                ;               20(SP)          INTERRUPT PC
                ;               24(SP)          INTERRUPT PSL
                ;
                ; OUTPUTS:
                ;
                ;       NONE
                ;
                ; SIDE EFFECTS:
                ;
                ;       INTERRUPTS ARE DISABLED ON THE DR32
                ;-


                ;
                ; DR32 DCR REGISTER DEFINITIONS
                ;-
                        $DEFINI DR
$DEF            DR_DCR          .BLKL   1       ; DR32 CONTROL REGISTER
                _VIELD  DR_DCR,0,<-
                        <ADPTYP,8>,-            ; ADAPTER TYPE
                        <ID2ERR,,M>,-           ; ID2 ERROR
                        <ID2TOS,2>,-            ; ID2 TIME-OUT STATUS
                        <,1>,-                  ; RESERVED
                        <ID1ERR,,M>,-           ; ID1 ERROR
                        <ID1TOS,2>,-            ; ID1 TIME-OUT STATUS
                        <RDS,,M>,-              ; READ DATA SUBSTITUTE
                        <CRD,,M>,-              ; CORRECTED READ DATA
                        <DCRHLT,,M>,-           ; DCR HALT
                        <DCRABT,,M>,-           ; DCR ABORT INTERRUPT
                        <PKTINT,,M>,-           ; PACKET INTERRUPT
                        <INTENB,,M>,-           ; INTERRUPT ENABLE
                        <,1>,-                  ; RESERVED
                        <PWR_UP,,M>,-           ; ADAPTER POWER UP
                        <PWR_DN,,M>,-           ; ADAPTER POWER DOWN
                        <EXTABT,,M>,-           ; EXTERNAL ABORT
                        <,1>,-                  ; RESERVED
                        <IMPDEP,6>,-            ; IMPLEMENTATION DEPENDENT BITS
                        >

                ; DCR CONTROL FIELD A CODES (USED WHEN WRITING TO DCR)

00000100                DCR_K_CLRPWRUP=^X100
```

```
                     00000200  0004   296          DCR_K_CLRPWRDN=^X200                    ; CLEAR POWER DOWN
                     00000300  0004   297          DCR_K_CLREXTABT=^X300                   ; CLEAR EXTERNAL ABORT
                     00000400  0004   298          DCR_K_CLRABTINT=^X400                   ; CLEAR ABORT INTERRUPT
                     00000500  0004   299          DCR_K_CLRINTENB=^X500                   ; CLEAR INTERRUPT ENABLE
                     00000600  0004   300          DCR_K_SETINTENB=^X600                   ; SET INTERRUPT ENABLE
                     00000700  0004   301          DCR_K_CLRHLT=^X700                      ; CLEAR HALT
                               0004   302
                               0004   303  ; DCR CONTROL FIELD B CODES (USED WHEN WRITING TO DCR)
                               0004   304
                     00001000  0004   305          DCR_K_CLRCRD=^X1000                     ; CLEAR CRD
                     00002000  0004   306          DCR_K_SETEXTABT=^X2000                  ; SET EXTERNAL ABORT
                     00003000  0004   307          DCR_K_CLRPKTINT=^X3000                  ; CLEAR PACKET INTERRUPT
                     00004000  0004   308          DCR_K_RESET=^X4000                      ; RESET
                     00005000  0004   309          DCR_K_SETOSQTST=^X5000                  ; SET OSEQ TEST
                     00006000  0004   310          DCR_K_CLROSQTST=^X6000                  ; CLEAR OSEQ TEST
                               0004   311          $DEFEND DR
                               0024   312
                               0024   313  DR$INT::
                        53 9E  D0      0024   314          MOVL    @(SP)+,R3               ; GET ADDRESS OF IDB
                        54 63  D0      0027   315          MOVL    IDB$L_CSR(R3),R4        ; GET ADDRESS OF FIRST CSR
                64 00000100 8F  D0     002A   316          MOVL    #DCR_K_CLRPWRUP,DR_DCR(R4)  ; CLEAR POWER UP
                64 00000200 8F  D0     0031   317          MOVL    #DCR_K_CLRPWRDN,DR_DCR(R4)  ; CLEAR POWER DOWN
                     52 8E  7D 0038   318          MOVQ    (SP)+,R2                ; RESTORE REGISTERS
                     54 8E  7D 003B   319          MOVQ    (SP)+,R4
                           02 003E   320          REI
                               003F   321
                               003F   324
                               003F   325  DR$INITIAL::                               ; CONTROLLER INITIALIZATION
                               003F   326  DR$SHUTDOWN::                              ; CONTROLLER SHUTDOWN
                               003F   327
                               003F   330
                  64 4000 8F   3C 003F   331          MOVZWL  #DCR_K_RESET,(R4)       ; RESET DR (R4 POINTS TO CSR)
                               0044   334
                           05  0044   335          RSB
```

```
                                0045   337          .SBTTL  UBA$INITIAL - CPU-DEPENDENT UNIBUS ADAPTER INITIALIZATION
                                0045   338 ;+
                                0045   339 ; UBA$INITIAL - UNIBUS ADAPTER INITIALIZATION
                                0045   340 ;
                                0045   341 ; THIS ROUTINE IS CALLED VIA A JSB INSTRUCTION AT SYSTEM STARTUP AND AFTER
                                0045   342 ; A POWER RECOVERY RESTART TO ALLOW INITIALIZATION OF UNIBUS ADAPTERS.
                                0045   343 ;       (POWERFAIL AND INITADP)
                                0045   344 ;
                                0045   345 ; INPUTS:
                                0045   346 ;
                                0045   347 ;       R2 = ADDRESS OF ADAPTER CONTROL BLOCK  (11/780 AND 11/750)
                                0045   348 ;       R4 = ADDRESS OF UNIBUS ADAPTER CONFIGURATION STATUS REGISTER (11/780)
                                0045   349 ;
                                0045   350 ;       ALL INTERRUPTS ARE LOCKED OUT.
                                0045   351 ;
                                0045   352 ; OUTPUTS:
                                0045   353 ;
                                0045   354 ;       THE UNIBUS ADAPTER IS INITIALIZED AND INTERRUPTS ARE ENABLED.
                                0045   355 ;-
                                0045   356
                                0045   357 UBA$INITIAL::                              ;UNIBUS ADAPTER INITIALIZATION
                                0045   358                                             ;
                                0045   360
          64    00    D2        0045   361          MCOML   #0,UBA$L_CSR(R4)           ;CLEAR ALL ADAPTER CONFIGURATION ERRORS
       08 A4    00    D2        0048   362          MCOML   #0,UBA$L_SR(R4)            ;CLEAR ALL ADAPTER STATUS BITS
    50  0256 C2    3C  004C   363          MOVZWL  ADP$W_UMR_DIS(R2),R0       ;PICK UP THE NUMBER OF UMR'S TO DISABLE
    50  50   16    78  0051   364          ASHL    #UBA$V_CR_MRDSB-4,R0,R0    ;DIVIDE BY 16 THEN SHIFT INTO POSITION
                    C9  0055   365          BISL3   #UBA$M_CR_SUEFIE!-         ;ENABLE INTERRUPTS
                                0056   366                   UBA$M_CR_BRIE!-
                                0056   367                   UBA$M_CR_CNFIE!-
                                0056   368                   UBA$M_CR_USEFIE!-
                                0056   369                   UBA$M_CR_IFSIE,-
04 A4   50   0000007C 8F       0056   370                   R0,UBA$L_CR(R4)
                                005E   371
                                005E   373
                                005E   383
                                005E   384 10$:                                       ;NO SPECIAL INIT FOR 11/730 OR uVAX I
                          05    005E   385          RSB                                ;
                                005F   386 ;
                                005F   387 ; IGNORE UNEXPECTED UNIBUS INTERRUPTS
                                005F   388 ;
                                005F   389
                                005F   390          .ALIGN  LONG
                                0060   391
                                0060   392 UBA$INTO::                                ; PASSIVE RELEASES THROUGH VECTOR 0
                                0060   393
          00000000'9F   D6      0060   394          INCL    a#IO$GL_UBA_INTO           ; COUNT THEM
                     00   11    0066   395          BRB     UBA_UNEXINT                ; JOIN COMMON CODE, VECTORS ARE ALLIGNED
                                0068   396
                                0068   397          .ALIGN  LONG
                                0068   398
                                0068   399 ;
                                0068   400 ; NOTE: UBA$UNEXINT is the label in the EXEC that is a JMP a#UBA_UNEXINT.
                                0068   401 ;       This seeming duplicity is necesary since there is code that must
                                0068   402 ;       refer to the EXEC address from within the SYSLOA image.
                                0068   403 ;
                                0068   404 UBA_UNEXINT::                             ; UNEXPECTED INTERRUPT CODE
```

```
                0068   405
                0068   407
                0068   408                                                  ; FOR 780-LIKE PROCESSORS, RESTORE
                0068   409                                                  ; SAVED REGISTERS
3F   BA         0068   409          POPR    #^M<R0,R1,R2,R3,R4,R5>       ;
                006A   412                                                  ; FOR 11/750, NO REGISTERS SAVED
02              006A   414          REI                                     ; IGNORE INTERRUPT
```

```
                                006B   418          .SBTTL  MASSBUS ADAPTER INTERRUPT DISPATCHER
                                006B   419 ;+
                                006B   420 ; MBASINT - MASSBUS ADAPTER INTERRUPT DISPATCHER
                                006B   421 ;
                                006B   422 ; THIS ROUTINE IS ENTERED VIA A JSB INSTRUCTION WHEN AN INTERRUPT OCCURS
                                006B   423 ; ON A MASSBUS ADAPTER. THE STATE OF THE STACK ON ENTRY IS:
                                006B   424 ;
                                006B   425 ;       00(SP) = ADDRESS OF IDB ADDRESS.
                                006B   426 ;       04(SP) = SAVED R2.
                                006B   427 ;       08(SP) = SAVED R3.
                                006B   428 ;       12(SP) = SAVED R4.
                                006B   429 ;       16(SP) = SAVED R5.
                                006B   430 ;       20(SP) = INTERRUPT PC.
                                006B   431 ;       24(SP) = INTERRUPT PSL.
                                006B   432 ;
                                006B   433 ; INTERRUPT DISPATCHING OCCURS AS FOLLOWS:
                                006B   434 ;
                                006B   435 ;       IF THE INTERRUPTING ADAPTER IS CURRENTLY OWNED AND THE OWNER UNIT
                                006B   436 ;       IS EXPECTING AN INTERRUPT, THEN THAT UNIT IS DISPATCHED FIRST. ALL
                                006B   437 ;       OTHER UNITS ARE DISPATCHED BY READING THE ATTENTION SUMMARY REG-
                                006B   438 ;       ISTER AND SCANNING FOR UNITS THAT HAVE ATTENTION SET. AS EACH UNIT
                                006B   439 ;       IS FOUND, ITS ATTENTION SUMMARY BIT IS CLEARED AND THEN A TEST IS
                                006B   440 ;       MADE TO DETERMINE IF AN INTERRUPT IS EXPECTED ON THE UNIT. IF YES,
                                006B   441 ;       THEN THE DRIVER IS CALLED AT ITS INTERRUPT RETURN ADDRESS. ELSE
                                006B   442 ;       THE DRIVER IS CALLED AT ITS UNSOLICITED INTERRUPT ADDRESS. AS EACH
                                006B   443 ;       CALL TO THE DRIVER RETURNS, THE ATTENTION SUMMARY REGISTER IS RE-
                                006B   444 ;       READ AND AN ATTEMPT IS MADE TO FIND ANOTHER UNIT TO DISPATCH. WHEN
                                006B   445 ;       NO UNITS REQUESTING ATTENTION REMAIN, THE INTERRUPT IS DISMISSED.
                                006B   446 ;-
                                006B   447
                                006B   448          .ALIGN  LONG
                                006C   449
                                006C   450 MBA$INT::                                ;MASSBUS ADAPTER INTERRUPT DISPATCHER
          53    00 BE    D0     006C   451          MOVL    @(SP),R3                ;GET ADDRESS OF IDB
            54    63    D0     0070   452          MOVL    IDB$L_CSR(R3),R4        ;GET ADDRESS OF CONFIGURATION STATUS REGISTE
                                0073   453
                                0073   455
          00800000 8F    D3     0073   456          BITL    #MBA$M_CSR_PD,-
                          64     0079   457                  MBA$L_CSR(R4)           ;CHECK FOR MBA POWER DOWN
                          61    12      007A   458          BNEQ    45$             ;BRANCH IF POWERFAIL
                                007C   459
                                007C   467
            55    04 A3    D0     007C   468          MOVL    IDB$L_OWNER(R3),R5      ;GET OWNER UNIT UCB ADDRESS
                          0A    13      0080   469          BEQL    10$             ;IF EQL NO OWNER
          52    0090 C5    9A     0082   470          MOVZBL  UCB$B_SLAVE(R5),R2      ;GET OWNER SLAVE CONTROLLER NUMBER
        21 64 A5    01    E0     0087   471          BBS     #UCB$V_INT,UCB$W_STS(R5),20$ ;IF SET, INTERRUPT EXPECTED
            53    00 BE    D0     008C   472 10$:     MOVL    @(SP),R3                ;RETRIEVE ADDRESS OF IDB
            54    63    D0     0090   473          MOVL    IDB$L_CSR(R3),R4        ;RETRIEVE MBA CONFIGURATION REGISTER ADDRESS
          08 A4    00    D2     0093   474          MCOML   #0,MBA$L_SR(R4)         ;CLEAR ALL MBA STATUS BITS
          52    0410 C4    D0     0097   475          MOVL    MBA$L_AS(R4),R2         ;READ ATTENTION SUMMARY REGISTER
    52    52    08    00    EA     009C   476          FFS     #0,#8,R2,R2             ;FIND FIRST UNIT REQUESTING ATTENTION
                          0A    12      00A1   477          BNEQ    20$             ;IF NEQ UNIT FOUND
                5E    04    C0     00A3   478          ADDL    #4,SP                   ;REMOVE IDB ADDRESS FROM STACK
                52    8E    7D     00A6   479          MOVQ    (SP)+,R2                ;RESTORE REGISTERS
                54    8E    7D     00A9   480          MOVQ    (SP)+,R4                ;
                                02     00AC   481          REI                     ;
                                00AD   482
```

```
                                                    H 3
                55   18 A342   D0  00AD   483 20$:    MOVL    IDB$L_UCBLST(R3)[R2],R5 ;GET ADDRESS OF UCB OR INTERRUPT DISPATCHER
                       22 55   E8  00B2   484          BLBS    R5,40$                  ;IF LBS INTERRUPT DISPATCHER FOR MULTI-
                               00B5   485                                              ; DEVICE CONTROLLER
          0410 C4    01   52   78  00B5   486          ASHL    R2,#1,MBA$L_AS(R4)      ;CLEAR ATTENTION SUMMARY BIT
                          55   D5  00B8   487          TSTL    R5                      ;SEE IF UCB DEFINED
                          CD   13  00BA   488          BEQL    10$                     ;IF EQL NONE DEFINED
             09 64 A5     01   E5  00BC   489          BBCC    #UCB$V_INT,UCB$W_STS(R5),30$ ;IF CLR, INTERRUPT NOT EXPECTED
                53   10 A5   7D  00C1   490          MOVQ    UCB$L_FR3(R5),R3        ;RESTORE DRIVER CONTEXT
                        0C B5   16  00C5   491          JSB     @UCB$L_FPC(R5)          ;CALL DRIVER AT INTERRUPT RETURN ADDRESS
                           BF   11  00C8   492          BRB     10$                     ;
                               00CA   493
                53   0088 C5   D0  00CA   494 30$:    MOVL    UCB$L_DDT(R5),R3        ;GET ADDRESS OF DDT
                        04 63   16  00CF   495          JSB     @DDT$L_UNSOLINT(R3)     ;CALL UNSOLICITED INTERRUPT ROUTINE
                           B5   11  00D2   496          BRB     10$                     ;
                               00D4   497
                           7E   DC  00D4   498 40$:    MOVPSL  -(SP)                   ;READ CURRENT PSL
                           75   16  00D6   499          JSB     -(R5)                   ;CALL SLAVE CONTROLLER INTERRUPT DISPATCHER
                           AF   11  00D8   500          BRB     10$                     ;
                               00DA   501
                               00DA   503
                               00DA   504 ;
                               00DA   505 ; IN CASE OF ADAPTER POWER DOWN BIT ASSERTED, RETRIEVE ADP ADDRESS AND JUMP
                               00DA   506 ; TO ADAPTER ERROR ROUTINE IN SYSLOA780.
                               00DA   507 ;
                               00DA   508
                54   14 A3     D0  00DA   509 45$:    MOVL    IDB$L_ADP(R3),R4        ;GET ADP ADDRESS
                       FF1C'    31  00DE   510          BRW     EXE$RA780_INT           ;JUMP TO ERROR ROUTINE
                               00E1   511
                               00E1   533
```

```
                     00E4    535          .SBTTL  MASSBUS ADAPTER INITIALIZATION
                     00E4    536 ;+
                     00E4    537 ; MBA$INITIAL - MASSBUS ADAPTER INITIALIZATION
                     00E4    538 ;
                     00E4    539 ; THIS ROUTINE IS CALLED VIA A JSB INSTRUCTION AT SYSTEM STARTUP AND AFTER
                     00E4    540 ; A POWER RECOVERY RESTART TO ALLOW INITIALIZATION OF MASSBUS ADAPTERS.
                     00E4    541 ;
                     00E4    542 ; INPUTS:
                     00E4    543 ;
                     00E4    544 ;       R4 = CSR ADDRESS OF MASSBUS ADAPTER.
                     00E4    545 ;       R5 = ADDRESS OF ADAPTER IDB.
                     00E4    546 ;
                     00E4    547 ;       ALL INTERRUPTS ARE LOCKED OUT.
                     00E4    548 ;
                     00E4    549 ; OUTPUTS:
                     00E4    550 ;
                     00E4    551 ;       THE MASSBUS ADAPTER IS INITIALIZED AND INTERRUPTS ARE ENABLED.
                     00E4    552 ;-
                     00E4    553
                     00E4    554 MBA$INITIAL::                               ;MASSBUS ADAPTER INITIALIZATION
                     00E4    555
   01     D0         00E4    558          MOVL    #MBA$M_CR_INIT,-
04 A4                00E6    559                   MBA$L_CR(R4)              ;INITIALIZE MASSBUS ADAPTER
   04     D0         00E8    560          MOVL    #MBA$M_CR_IE,-
04 A4                00EA    561                   MBA$L_CR(R4)              ;ENABLE INTERRUPTS
                     00EC    564
                05   00EC    565          RSB
```

```
                        00ED   567          .SBTTL  INI$MPMADP - BUILD ADP AND INITIALIZE MULTI-PORT MEMORY
                        00ED   568 ;+
                        00ED   569 ; INI$MPMADP IS CALLED AFTER  MAPPING THE REGISTERS FOR A MULTI-PORT
                        00ED   570 ; MEMORY ADAPTER.  AN ADAPTER CONTROL BLOCK IS ALLOCATED AND FILLED.
                        00ED   571 ; THE HARDWARE ADAPTER IS THEN INITIALIZED BY CALLING MPM$INITIAL.
                        00ED   572 ;
                        00ED   573 ; NOTE: THIS ROUTINE HAS BEEN LOCATED HERE IN SYSLOAXXX.EXE INSTEAD OF
                        00ED   574 ;       INILOA.EXE BECAUSE IT CAN BE CALLED WHILE THE SYSTEM IS RUNNING
                        00ED   575 ;       LONG AFTER INILOA.EXE HAS BEEN DELETED!!!
                        00ED   576 ;
                        00ED   577 ; INPUT:
                        00ED   578 ;       R4 - nexus identification number of this nexus
                        00ED   579 ;
                        00ED   580 ; OUTPUTS:
                        00ED   581 ;       ALL REGISTERS PRESERVED
                        00ED   582 ;-
                        00ED   583
                00000010 00ED  584 NUMMPMVEC = 16                          ; NUMBER OF INTER-PORT INTERRUPT VECTORS
                        00ED   585
                        00ED   586 INI$MPMADP::                            ; INITIALIZE MPM DATA STRUCTURES
                        00ED   587
                     05 00ED   589          RSB                            ; DUMMY ENTRY FOR SYSGEN
                        00EE   590
```

```
                                .SBTTL  MASINITIAL - INITIALIZE MULTI-PORT MEMORY ADAPTER
                        ;++
                        ;
                        ; MPM$INITIAL - INITIALIZE MULTI-PORT MEMORY ADAPTER
                        ;
                        ; THIS ROUTINE IS CALLED AT SYSTEM INTIALIZATION AND AFTER A POWER
                        ; RECOVERY RESTART TO INITIALIZE THE PORT ADAPTER BY CLEARING ANY
                        ; ERRORS AND ENABLING ALL INTERRUPTS.
                        ;
                        ; INPUTS:
                        ;
                        ;       R4 = ADDR OF ADAPTER CSR.
                        ;
                        ;       IPL = 31
                        ;
                        ; OUPUTS:
                        ;
                        ;       ANY ERRORS IN PORT ARE CLEARED AND ALL INTERRUPTS ARE ENABLED.
                        ;--
                        
                        MASINITIAL::                            ; INTIALIZE PORT
                        
05                              RSB
                        
                        
```

```
                        .SBTTL  INTER-PROCESSOR REQUEST HANDLER
                ;++
                ;
                ; FUNCTIONAL DESCRIPTION:
                ;
                ;       THIS ROUTINE IS CALLED BY A DRIVER OR AN EXEC FUNCTION TO
                ;       EITHER SEND A REQUEST TO OR JUST INTERRUPT ANOTHER PROCESSOR
                ;       THAT IS CONNECTED TO A PORT OF THE MULTIPORT MEMORY.
                ;
                ; INPUTS:
                ;
                ;       R4 = ADAPTER CONTROL BLOCK ADDRESS.
                ;       R5 = IF LSS 0 - ADDRESS OF A FORK BLOCK TO USE IF REQUEST
                ;                       BLOCK IS NOT AVAILABLE.
                ;            IF GEQ 0 - PORT NUMBER OF PROCESSOR TO JUST INTERRUPT.
                ;
                ; OUTPUTS:
                ;
                ;       WHEN THIS ROUTINE IS CALLED WITH A FORK BLOCK ADDRESS, IT WILL
                ;       ATTEMPT TO ALLOCATE A REQUEST BLOCK.  IF THE REQUEST FAILS,
                ;       THE CONTEXT OF THE CALLER WILL BE SAVED IN THE FORK BLOCK, THE
                ;       FORK BLOCK BLOCK WILL BE INSERTED IN THE REQUEST BLOCK WAIT
                ;       QUEUE AND A RETURN TO THE CALLER'S CALLER IS EXECUTED.
                ;
                ;       IF A REQUEST BLOCK IS ALLOCATED SUCCESSFULLY, CONTROL WILL
                ;       RETURN TO THE CALLER VIA A CO-ROUTINE CALL SO THE CALLER CAN
                ;       FILL-IN THE REQUEST BLOCK.
                ;
                ;       THE CALLER WILL THEN PERFORM ANOTHER CO-ROUTINE CALL TO RETURN
                ;       TO THIS ROUTINE SO THE BLOCK CAN BE INSERTED IN THE DESIRED
                ;       PROCESSOR'S INTER-PROCESSOR REQUEST QUEUE.  IF IT IS THE
                ;       FIRST REQUEST IN THE QUEUE AN INTER-PORT INTERRUPT WILL
                ;       ALSO BE REQUESTED TO WAKE-UP THE DISPATCHER ON THE PORT.
                ;
                ;
                ;       IF THIS ROUTINE IS CALLED WITH A PORT NUMBER INSTEAD OF A
                ;       FORK BLOCK ADDRESS, IT WILL JUST REQUEST AN INTERRUPT FOR
                ;       THE PROCESSOR ON THE SPECIFIED PORT.  IT IS THEN UP TO THE
                ;       INTERRUPTED PROCESSOR TO DETERMINE WHAT THE INTERRUPT WAS
                ;       FOR.
                ;
                ;       R0 = SUCCESS OR FAILURE OF OPERATION.  THIS SHOULD BE CHECKED
                ;            BY THE CALLER BOTH TIMES THIS ROUTINE RETURNS.
                ;
                ;       R3,R4,R5 ARE PRESERVED.
                ;
                ;--
                ;
MA$REQUEST::                                    ; REQUEST HANDLER
                ;
05              RSB
```

```
                     .SBTTL  REPORT RESOURCE AVAILABILITY TO INTERESTED PORTS
                ;++
                ;
                ; FUNCTIONAL DESCRIPTION:
                ;
                ;       THIS ROUTINE IS CALLED TO REPORT TO ANY PROCESSORS THAT A RESOURCE
                ;       HAS BEEN MADE AVAILABLE.
                ;
                ; INPUTS:
                ;
                ;       R0 = RESOURCE NUMBER OF RESOURCE MADE AVAILABLE.
                ;       R1 = SHARED MEMORY CONTROL BLOCK (SHB) ADDRESS.
                ;
                ; OUTPUTS:
                ;
                ;       ANY PROCESSORS WAITING FOR THE SPECIFIED RESOURCE ARE INTERRUPTED
                ;       TO NOTIFY THEM THE RESOURCE IS AVAILABLE.
                ;
                ;       R0,R1,R2,R3 ARE NOT PRESERVED.
                ;--

                MA$RAVAIL::

05              RSB

                .END
```

```
ADP$W_UMR_DIS                   = 00000256
C780_CIKE                       = 00000001
CI$INITIAL                        0000001F RG    02
CI$INT                            00000000 RG    02
CI$SHUTDOWN                       0000001F RG    02
CPU_TYPE                        = 00000004
DCR_K_CLRPWRDN                  = 00000200
DCR_K_CLRPWRUP                  = 00000100
DCR_K_RESET                     = 00004000
DDT$L_UNSOLINT                  = 00000004
DR$INITIAL                        0000003F RG    02
DR$INT                            00000024 RG    02
DR$SHUTDOWN                       0000003F RG    02
DR_DCR                            00000000
EXE$RH780_INT                     ******** X   02
IDB$L_ADP                       = 00000014
IDB$L_CSR                       = 00000000
IDB$L_OWNER                     = 00000004
IDB$L_UCBLST                    = 00000018
INI$MPMADP                        000000ED RG    02
IO$GL_UBA_INTO                    ******** X   02
MA$INITIAL                        000000EE RG    02
MA$RAVAIL                         000000F0 RG    02
MA$REQUEST                        000000EF RG    02
MBA$INITIAL                       000000E4 RG    02
MBA$INT                           0000006C RG    02
MBA$L_AS                        = 00000410
MBA$L_CR                        = 00000004
MBA$L_CSR                       = 00000000
MBA$L_SR                        = 00000008
MBA$M_CR_IE                     = 00000004
MBA$M_CR_INIT                   = 00000001
MBA$M_CSR_PD                    = 00800000
NUMMPMVEC                       = 00000010
PA_CNF                            00000000
PA_CNF_M_PDN                    = 00800000
PA_CNF_M_PUP                    = 00400000
PA_PMC                            00000004
PA_PMC_M_MIN                    = 00000001
PR$_SID_TYP730                  = 00000003
PR$_SID_TYP750                  = 00000002
PR$_SID_TYP780                  = 00000001
PR$_SID_TYP790                  = 00000004
PR$_SID_TYPUV1                  = 00000007
SIZ...                          = 00000006
UBA$INITIAL                       00000045 RG    02
UBA$INTO                          00000060 RG    02
UBA$L_CR                        = 00000004
UBA$L_CSR                       = 00000000
UBA$L_SR                        = 00000008
UBA$M_CR_BRIE                   = 00000020
UBA$M_CR_CNFIE                  = 00000004
UBA$M_CR_IFSIE                  = 00000040
UBA$M_CR_SUEFIE                 = 00000008
UBA$M_CR_USEFIE                 = 00000010
UBA$V_CR_MRDSB                  = 0000001A
UBA_UNEXINT                       00000068 RG    02
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
ABS	00000008 (8.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
SYSLOA	000000F1 (241.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.05	00:00:02.13
Command processing	129	00:00:00.57	00:00:03.97
Pass 1	541	00:00:13.90	00:00:53.79
Symbol table sort	0	00:00:02.21	00:00:07.70
Pass 2	113	00:00:02.88	00:00:10.56
Symbol table output	8	00:00:00.08	00:00:00.29
Psect synopsis output	2	00:00:00.01	00:00:00.53
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	824	00:00:19.70	00:01:18.97

The working set limit was 1950 pages.
131956 bytes (258 pages) of virtual memory were used to buffer the intermediate code.
There were 110 pages of symbol table space allocated to hold 2138 non-local and 6 local symbols.
1179 source lines were read in Pass 1, producing 13 object records in Pass 2.
38 pages of virtual memory were used to define 37 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYSLOA.OBJ]790DEF.MLB;1	0
_255DUA28:[SYS.OBJ]LIB.MLB;1	25
_255DUA28:[SYSLIB]STARLET.MLB;2	7
TOTALS (all libraries)	32

2215 GETS were required to define 32 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:ADPSUB790/OBJ=OBJ$:ADPSUB790 MSRC$:CPUSW790/UPDATE=(ENH$:CPUSW790)+MSRC$:ADPSUB/UPDATE=(ENH$:ADPSUB)+EXECML$/LIB+LIB$

CNXOPT
LIS

ADPSUBUV1
LIS

CNXMAN
LIS

CJFCLUSTR
LIS

CLUSTRLOA
LIS

CONMAN
LIS

ADPSUB790
LIS

CLUMESSAG
LIS